

Optimizing products and their production processes by neural analysis

Patent Number: DE19742906
Publication date: 1999-05-06
Inventor(s): OTTE VIKTOR PROF DR ING (DE); OTTE RALF DIPL ING (DE)
Applicant(s):: ABB PATENT GMBH (DE)
Requested Patent: ☐ DE19742906
Application Number: DE19971042906 19970929
Priority Number(s): DE19971042906 19970929
IPC Classification: G06F17/50 ; G05B19/4097 ; B62D65/00 ; C07B61/00 ; C12N15/00 ; B01J19/00 ; F01K13/02
EC Classification: B01J19/00B
Equivalents:

Abstract

Product and/or process optimization is carried out by neural analysis based on self-organizing maps (SOMs). A process for optimization problem solving in research, development and construction and for optimization of technical or chemical products and their production processes comprises collecting and processing all the values relevant to the optimization by a neural analysis based on self-organizing maps (SOMs), in which a topology producing, non-linear projection of data of the relevant parameter and the associated target values is performed on a multi-dimensional SOM. The target value is optimized by displaying as a SOM component map in either height coded or color coded form and, after selecting values for the target value, the underlying parameter combinations are calculated and output.

Data supplied from the **esp@cenet** database - I2

THIS PAGE BLANK (USPTO)



- 71 Anmelder:
ABB Patent GmbH, 68309 Mannheim, DE
- 72 Erfinder:
Otte, Viktor, Prof. Dr.-Ing.habil., 42657 Solingen, DE;
Otte, Ralf, Dipl.-Ing., 69469 Weinheim, DE
- 56 Entgegenhaltungen:
DE 196 37 651 A1
DE 44 43 192 A1
DE 40 29 774 A1
DE 37 10 990 A1
WENZEL,Lothar: Optimalstrategie. In: mc extra,
10/95, S.30-33;
HAMPEL,Rainer, CHAKER,Nasredin:
Dampfturbinen-
regelung mit Fuzzy-Logik. In: atp -
Automatisierungstechnische Praxis 37, 1995,
6, S.32-41;
GEHLEN,Stefan, u.a.: Einsatz Neuronaler Netze
zur Kontrolle komplexer industrieller Prozesse.
In: at - Automatisierungstechnik 43, 1995, 2,
S.85-91;
GIEREND,Chr.: Feuerungsregelung durch Fuzzy
Control. In: VGB Kraftwerkstechnik 76, 1996,

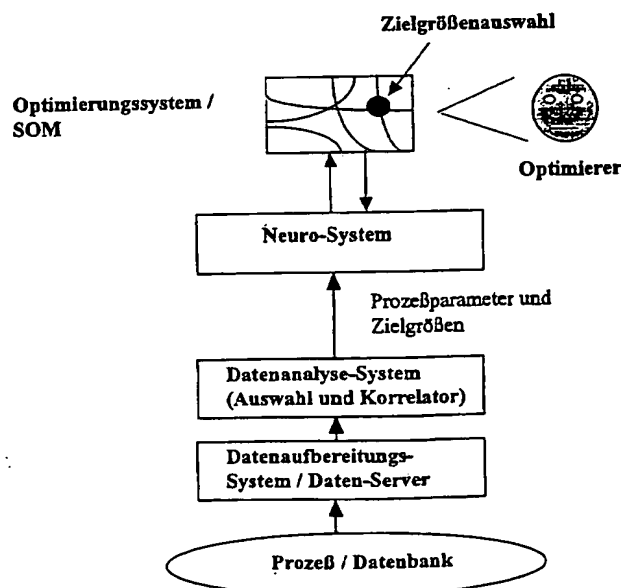
H.6, S.487-497;
Prozeßmodelle auf Basis Neuronaler Netze. In:
atp 4/97, S.66;
SPUR,Günter, KRAUSE,Frank-Lothar: CAD-Technik,
Carl Hanser Verlag, München, Wien, 1984, S.282-
S.285;
ROJAS,Paúl: Theorie der neuronalen Netze,
Springer-Verlag, Berlin u.a., 1993, S.339-359;
BRAUSE,Rüdiger: Neuronale Netze, B.G.Teubner,
Stuttgart, 1995, 2.Aufl., S.152-164;
Prospekt: IDL Basics, RSI Research Systems, Inc.,
Nov. 92, S.13-16,30,31,46-55;
KOHONEN,Teuvo, et.al.: Engineering Applications
of the Self-Organizing Map. In: Proceedings Of
The IEEE, Vol.84, No.10, Oct. 1996, S.1358-1382;
HALICI,Ugur, ONGUN,Güçlü: Fingerprint
Classification Through Self-Organizing Feature
Maps Modified to Treat Uncertainties. In:
Proceedings Of the IEEE, Vol.84, No.10,
Oct. 1996, S.1497-1512;
GRAEDEL,T.E., MCGILL,R.: Graphical Presentation
of Results from Scientific Computer Models. In:
SCIENCE, Vol.215, 5. March 1982, S.1191-1198;
BAUER, H.-U., u.a.: Selbstorganisierende Karten,
in: Spektrum der Wissenschaften, April 1996,
S. 38-47;

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

54 Verfahren zum Optimieren von Produkten und Produktionsprozessen

57 Die Erfindung bezieht sich auf ein Verfahren zur Lösung allgemeiner Optimierungsaufgaben in Forschung, Entwicklung und Konstruktion und zum Optimieren von technischen oder chemischen Produkten und von Prozessen zur Herstellung dieser Produkte. Um eine gleichzeitige und zusammenhängende Bewertung relevanter Parameter und der dazugehörigen Zielgrößen zu ermöglichen, wird vorgeschlagen, die Einflüsse der Prozeßgrößen durch eine neuronale Analyse auf der Grundlage selbstorganisierender Karten in Beziehung zueinander auszuwerten, indem eine topologieerhaltende, nichtlineare Projektion von Daten der Versuchsparameter auf eine 2-dimensionale neuronale Karte (sogenannter SOM) realisiert wird. Nach einer Auswahl von Werten der Zielgröße (Z1, Z2) auf der SOM-Komponentenkarte werden die zugrundeliegenden Parameterkombinationen berechnet und ausgegeben.



Beschreibung

Die Erfindung bezieht sich auf ein Verfahren zur Lösung von Optimierungsaufgaben in Forschung, Entwicklung und Konstruktion und zum Optimieren von technischen oder chemischen Produkten und von Prozessen zur Herstellung dieser Produkte.

Das Verfahren ist geeignet zum Optimieren von Konstruktionsprozessen im Maschinen und Fahrzeugbau, in der Feinwerktechnik bzw. bei Entwicklungsprozessen für gentechnische Produkte und zur Optimierung von Anlagen oder Anlagenteilen eines Kraftwerks.

Es ist allgemein bekannt, für Optimierungsprobleme numerische Methoden der Parameteroptimierung anzuwenden. Dabei geht man von einem Satz N unabhängiger Variablen p_1, p_2, \dots, p_N aus, die in einem sogenannten Parametervektor p zusammengefaßt werden:

$$p = (p_1, p_2, \dots, p_N) \quad (1)$$

Die Qualität eines jeden solchen Parametervektors wird durch eine Gütefunktion Q beschrieben. $Q(p)$ sagt damit aus, wie gut der gewählte Parametervektor p das Optimierungsproblem löst. Im allgemeinen wird Q so definiert, daß das Optimierungsproblem um so besser gelöst wird, je größer oder kleiner Q ist. Geht man davon aus, daß ein minimales Q das Optimum darstellt, wird also

$$Q(p) \rightarrow \text{Minimum} \quad (2)$$

gesucht. Für diese Aufgabe sind mehrere Lösungsmethoden bekannt, die man grob in deterministische und in Zufallsverfahren aufteilen kann. Zahlreiche deterministische Verfahren, z. B. Gradientenverfahren und Newton-Verfahren, haben den praxisrelevanten Nachteil, daß für deren Lösung partielle Ableitungen der Gütefunktion zur Verfügung stehen müssen. Andere deterministische Verfahren, z. B. direkte Suchverfahren, bleiben oftmals in den lokalen Extrema ihrer Umgebung stecken und finden nicht das gewünschte globale Optimum, siehe J. Kahlert, "Fuzzy Control für Ingenieure", Vieweg Verlag, 1995, Seiten 155 bis 170. Eine Stärke der klassischen Optimierungsverfahren besteht – wenn sie anwendbar sind – darin, daß sie das Optimum finden können, ohne über alle Bereiche Informationen besitzen zu müssen. Dagegen fehlt ihnen die Universalität in der Anwendbarkeit, da mathematische Modelle vorausgesetzt werden.

Weiterhin ist bekannt, daß es gerade in Forschung, Entwicklung und Konstruktion zu Optimierungsaufgaben kommt, die nicht durch die Spezifikation eines einzigen Gütekriteriums gelöst werden können, sondern die aus gleichzeitig zu optimierenden Zielfunktionen bestehen, die oftmals entgegengesetzte Anforderungen an den Prozeß stellen. Damit entsteht eine sogenannte vektorielle Gütefunktion und der Zielkonflikt wird durch Methoden der Polyoptimierung formal behandelt, in deren Ergebnis die beste Kompromißlösung gesucht wird. Damit verschärfen sich die obengenannten Probleme der existierenden Optimierungsverfahren nochmals, so daß diese in vielen praktischen Anwendungsfällen nicht eingesetzt werden können.

Es ist weiterhin bekannt, daß zufallsgesteuerte Optimierungsverfahren, z. B. Monte-Carlo-Verfahren oder Evolutionsstrategien, obengenannte Nachteile überwinden können; deshalb sind diese in zahlreiche praktische Anwendungen integriert worden. Sie gehen davon aus, daß die skalare oder vektorielle Gütefunktion in dem Parameterraum P ein mehrdimensionales Gebirge darstellt, dessen Minima respektive Maxima gefunden werden sollen, siehe E. Schöneburg et. al.

"Genetische Algorithmen und Evolutionsstrategien", Addison Wesley Verlag, 1994, Seiten 102 bis 107 und die Seiten 141 bis 215.

Ein Nachteil dieser zufallsgesteuerten Optimierungsverfahren ist, daß man weiterhin keine oder nur wenige Information über den Aufbau und die Topologie der Gütefunktionen Q im Zustandsraum P des Optimierungsproblems hat. In der Systemtheorie wird als Zustandsraum P eines Prozesses der n -dimensionale Vektorraum bezeichnet, in dem die zu untersuchenden Daten übereinander aufgetragen werden können. Zum Beispiel entspricht ein Datenbeispiel p einem Punkt in diesem Zustandsraum. Nachstehend wird unter einem Eingangsraum oder Zustandsraum P genau dieser Zustandsraum verstanden.

Weiterhin weiß man nicht, ob das gefundene Optimum lokal oder global ist. Ein weiterer Nachteil liegt in der richtigen Parametrisierung des Optimierungsverfahrens selbst, da z. B. die anfänglich einfachen Konzepte der Evolutionsstrategien oder Genetischen Algorithmen durch adaptive Schrittweltensteuerung, verschiedene Crossing-Over-Strategien oder Selektionsvarianten für die Anwendung im technischen oder chemischen Prozeß oftmals zu komplex geworden sind (siehe E. Schöneburg et al.).

Der Erfindung liegt die Aufgabe zugrunde, ein auf Daten basiertes, universelles Optimierungsverfahren anzugeben, das eine gleichzeitige und zusammenhängende Bewertung und Visualisierung aller relevanten Optimierungsparameter und Zielgrößen eines technischen oder chemischen Produktes und des Prozesses zur Herstellung dieses Produktes ermöglicht und die Gütefunktionen der Optimierung grafisch anschaulich darstellen kann.

Diese Aufgabe wird gelöst durch ein im Anspruch 1 angegebenes Verfahren zur Optimierung von technischen oder chemischen Produkten und zur Optimierung von Prozessen in Forschung, Entwicklung, Konstruktion und Produktion dieser Produkte. Dabei werden alle für einen Prozeß relevanten Parameter zusammengefaßt und durch eine neuronale Analyse auf der Grundlage selbstorganisierender Karten (Self-Organizing-Map, sogenannte SOM) in Beziehung zueinander ausgewertet, indem eine topologieerhaltende, nichtlineare Projektion von Daten, d. h. von allen Parametern und Zielgrößen, des relevanten Prozesses auf eine mehrdimensionale, höhen- oder farbkodierte, neuronale Karte (SOM) realisiert wird. Da auf dieser strukturierten SOM die Zielgrößen in Form von sogenannten Komponentenkarten höhen- oder farbkodiert aufgetragen werden können, erhält man Informationen über die Zielgrößenverteilung im gesamten Parameterraum P . Damit können die gesuchten Werte der Zielgröße, d. h. die Minima, Maxima oder auch bestimmte Bereiche, selektiert und die dazugehörigen Parameterkombinationen ermittelt werden.

Die einzige Voraussetzung für das Verfahren ist das Vorhandensein von Beispieldaten für das Optimierungsproblem. Das Verfahren stellt damit eine Erweiterung bestehender Optimierungsmethoden dar.

Die Gütefunktion für die Optimierung kann eine Zielgröße selbst sein oder aus einer oder mehreren Zielgrößen berechnet werden. Ausgehend von der grafischen Darstellung des im allgemeinen hochdimensionalen Gütegebirges erhält man Informationen über die Topologie der Gütefunktionen und es können die globalen Minima und Maxima aus dem Gebirge einfach ausgelesen und die dazugehörigen Parameterkombinationen ausgegeben werden. Weiterführend sollen einzelne skalare Zielgrößenfunktionen, d. h. einzelne Zielgrößengebirge, miteinander überlagert werden können, so daß eine grafisch orientierte Polyoptimierung durchgeführt werden kann.

Vorteilhafte Ausgestaltungen des Verfahrens sind in wei-

teren Ansprüchen angegeben. Durch den ganzheitlichen Ansatz werden nicht nur die Werte einzelner Parameter, sondern auch gerade ihre gegenseitigen und nichtlinearen Einflüsse untereinander berücksichtigt.

In der weiteren Beschreibung wird unter einem Prozeß jede Operation verstanden, die das Ziel hat etwas herzustellen, zu verbessern oder zu optimieren. Ein Prozeß kann damit ein technischer oder chemischer Prozeß selbst, aber auch ein Vorgang in Forschung, Entwicklung und Konstruktion zur Herstellung eines Produktes bzw. zur Optimierung von Produkteigenschaften sein. Prozeßgrößen sind die den Prozeß beschreibenden und für die Optimierungsaufgabe relevanten Größen, z. B. Konstruktionsparameter, Produkteigenschaften oder physikalische Meßwerte.

Ein Prozeß im oben genannten Sinne wird durch folgenden Vektor $pvec$ von Prozeßgrößen beschrieben:

$$pvec = (p_1, p_2, \dots, p_i, \dots, p_N, z_1, z_2, \dots, z_j, \dots, z_L) \quad (3)$$

wobei p_i die i -te Einflußgröße oder auch der i -te Prozeßparameter und z_j die j -te Zielgröße des Prozesses ist, d. h. die Prozeßgrößen bestehen aus Parametern und Zielgrößen. Die Anzahl $n = N + L$ gibt die Gesamtanzahl der zugrundeliegenden Größen für den Prozeß an. N ist die Anzahl der Einflußgrößen, L ist die Anzahl der Zielgrößen; n ist damit die Anzahl aller relevanten Produkt- oder Prozeßgrößen. Als Datenvektor wird eine konkrete Ausprägung von $pvec_i$ verstanden, die zu einem Zeitpunkt t_i erfaßt wurde. Aufbauend auf vorhandenen Datenvektoren aus diesem Prozeß wird das SOM-Verfahren angewendet.

In der Systemtheorie wird als Zustandsraum R eines Prozesses der n -dimensionale Vektorraum bezeichnet, in dem die Prozeßdaten einander zugeordnet aufgetragen werden können. Ist R so gestaltet, daß jeder Größe eine Achse in R zugeordnet wird, entsteht für jedes Datenbeispiel $pvec_i$ ein Punkt in R . Ein solches Koordinatensystem R wird als Zustandsraum des Prozesses bezeichnet und jeder Prozeßzustand $pvec_i$ zum Zeitpunkt t_i als Punkt in R markiert. Nachstehend wird unter einem Eingangsraum genau dieser Zustandsraum verstanden. Der Ausgangsraum ist der 2-dimensionale Raum, der auf der SOM-Karte abgebildet wird. Jedem Vektor $pvec_i$ entspricht somit ein Punkt in R , eine Reihe von Vektoren stellt eine Punktwolke in R dar. Aufbauend auf zahlreichen Daten für das Optimierungsproblem und damit auf einer oben genannten Punktwolke wird das SOM-Verfahren angewendet.

In der neuronalen Theorie versteht man unter einer selbstorganisierenden Karte (SOM) ein "selbstorganisierendes neuronales Netz", bei dem alle Neuronen nebeneinander angeordnet sind. Das selbstorganisierende neuronale Netz ist ein eingeführter Begriff für eine spezielle Klasse von neuronalen Netzen, die sich anhand von Eingangssignalen selbst strukturieren, vgl. A. Zell "Simulation Neuronaler Netze", Addison-Wesley Verlag, 1994, Seite 179 bis 187. Im Unterschied zu herkömmlichen neuronalen Netzen spielt die räumliche Lage der einzelnen Neuronen und ihre Nachbarschaftsbeziehungen bei den SOM eine wichtige Rolle. Die Begriffe SOM und Karte werden im weiteren gleichrangig verwendet.

Mit Hilfe der SOMs lassen sich sogenannte topologieerhaltende Abbildungen realisieren. Topologieerhaltend heißt in diesem Zusammenhang, daß die Punkte (Datenpunkte), die im Eingangsraum nahe beieinander liegen, auch im Ausgangsraum, also auf der Karte, nahe beieinander liegen werden. Damit stellt die Karte im Prinzip ein topologieerhaltendes, 2-dimensionales Fenster in den n -dimensionalen Zustandsraum des Prozesses dar.

Beim erfindungsgemäßen Verfahren werden nach einer

entsprechenden Datenvorverarbeitung der Prozeßgrößen einem selbstorganisierenden Netz in einer Lernphase die Werte der für die Optimierungsaufgabe relevanten n Größen angeboten. Die Anzahl der vorliegenden Datenbeispiele sollte eine Mindestanzahl (MA) nicht unterschreiten. Diese Anzahl ist:

$$MA = 30 \cdot n \quad (4)$$

Das Verfahren arbeitet auch mit weniger Datenbeispielen, allerdings werden dadurch die Ergebnisse verschlechtert. Der Wert von n kann zwischen 2 und mehreren hundert liegen. Unter Prozeß- oder Meßgröße werden beides, die Einflußparameter mit der Anzahl N und die Zielgrößen des Prozesses mit der Anzahl L verstanden. Die Anzahl der Zielgrößen sollte allerdings deutlich kleiner als die Anzahl der Einflußgrößen sein:

$$L \ll N. \quad (5)$$

Die Lernphase erfolgt in zwei Schritten: Zuerst entfaltet sich die Karte im Zustandsraum des Prozesses, danach werden die Datenbeispiele durch die Anwendung eines mathematischen Verfahrens visualisiert.

Bevor das Lernverfahren angewendet wird, werden die Prozeßgrößen vorverarbeitet. Zuerst werden die einzelnen Größen auf einen vorgegebenen Wert skaliert und danach z. B. noch verrauscht. Das Verrauschen erlaubt das Vervielfältigen und weitere Verarbeiten von typisierten Optimierungsbeispielen, da dadurch einzelne Beispieldaten, die stets einem systematischen und statistischen Fehler unterliegen, verallgemeinert werden. Der Rauschanteil darf dabei allerdings nicht so groß sein, daß er bestehende Datenverteilungen qualitativ verändert.

Die Entfaltung der selbstorganisierenden Karte im Zustandsraum des Prozesses wird durch Algorithmen, basierend auf einem neuronalen Algorithmus nach T. Kohonen realisiert. Der "selbstorganisierende neuronale SOM-Algorithmus" wurde von Kohonen im Jahr 1982 eingeführt; vgl. T. Kohonen, "Self-organized formation of topologically correct feature maps" in Biological Cybernetics, 43, 59 bis 69, 1982. Dabei wird jedem Neuron M auf der SOM ein n -dimensionaler Gewichtsvektor zugeordnet, der durch Anwendung des Verfahrens iterativ adaptiert wird. Jedes Neuron M hat demnach einen Gewichtsvektor w^M mit n Elementen:

$$w^M = (w_1^M, w_2^M, w_3^M, \dots, w_i^M, \dots, w_n^M). \quad (6)$$

Die Anzahl der Elemente oder Komponenten von w entspricht damit der Anzahl der zu untersuchenden Größen n , mit $n = N + L$ in Gleichung (3). Die Anzahl der Neuronen einer SOM wird mit k bezeichnet. Ein entfaltetes SOM-Netz wird i. a. als rechteckige Karte mit $x \cdot y = k$ Neuronen dargestellt, x bezeichnet die Anzahl der Neuronen in x -Richtung, y die Anzahl in y -Richtung. M ist ein ausgewähltes Neuron auf der SOM, mit $M = 1 \dots k$.

Durch den Algorithmus werden die Gewichtsvektoren w aller Neuronen k im Zustandsraum R verteilt. Die SOM kann sich auf jedem n -dimensionalen Gebilde (z. B. Kurve, Fläche oder Körper) selbstorganisieren, allerdings läßt sich nur durch eine erfindungsgemäße Modifikation des Verfahrens eine anwendbare und reproduzierbare Analyse der Prozeßdaten realisieren. Und zwar werden zwei ausgewählte gegenüberliegende Eckneuronen auf der rechteckigen SOM-Karte M_1 und M_2 auf das Minimum bzw. das Maximum aller Prozeßgrößen in P festgelegt und dort mathematisch fixiert, indem die Gewichtsvektoren dieser Eckneuronen w^{M_1} und w^{M_2} mit den Vektoren für die skalierten Minima und

Maxima der Prozeßgrößen gleichgesetzt werden. Minimum heißt damit am nächsten am Koordinatenursprung in R, Maximum am weitesten von diesem entfernt. Dadurch wird sich die Karte reproduzierbar entfalten, da ein Drehen der Karte in R verhindert werden kann.

Die Visualisierung der Situation im Zustandsraum R geschieht nach der Umatrix-Methode oder einem dynamischen Visualisierungsverfahren.

Bei der Umatrix-Methode werden die Differenzen der Gewichtsvektoren eines jeden Neurons zu seinen Nachbarn berechnet und entsprechend graphisch aufbereitet, z. B. farbkodiert dargestellt. Zusammenhängende Gebiete (z. B. helle Flächen) haben eine kleine Differenz, da ihre Neuronen im Zustandsraum nahe beieinander platziert wurden, zwischen diesen einzelnen Clustern gibt es Grenzen, die durch eine hohe Differenz der jeweiligen Gewichtsvektoren gekennzeichnet sind, diese werden z. B. als dunkle Flächen visualisiert. Damit läßt sich eine gute Visualisierung aller Clustergrenzen und damit der realen Prozeßzustände erreichen, siehe auch G. Whittington, C. Spracklen: The Application of Neural Network Mode to Sensor Data Fusion in Proc. of Applications of ANN Conference, Orlando, USA, 1990.

Bei den dynamischen Verfahren werden der entfalten Karte die Eingangsvektoren in zufälliger Reihenfolge nochmals angeboten und das aktuelle Gewinnerneuron nach einem "Winner-takes-all" Algorithmus ermittelt. Dieser Algorithmus besagt, daß das Neuron dessen Gewichtsvektor dem Eingangsvektor am nächsten ist gewinnt. Die Aussage "am nächsten" wird durch Berechnung eines vorher bestimmten Abstandsmaßes, z. B. dem euklidischen Abstand, berechnet. Der "Winner-takes-all" Algorithmus ist ein Verfahren, bei dem immer das Neuron ausgewählt wird, das eine gewisse Bedingung am besten erfüllt, alle anderen Neuronen des Netzes oder der Karte sind inaktiv bzw. nicht ausgewählt (1-aus-k Auswahl mit k ist gleich Anzahl Neuronen auf der Karte). Diese besondere Bedingung ist hier ein minimales Abstandsmaß zwischen dem aktuellen Eingangsvektor und dem Gewichtsvektor der einzelnen Neuronen, siehe S. Hafner, "Neuronale Netze in der Automatisierungstechnik", Oldenbourg Verlag, 1994, insbesondere Seiten 17 bis 25.

Für das jeweilige Gewinnerneuron wird ein interner Zähler, die Gewinnerrate, um den Zahlenwert eins erhöht. Am Ende dieser dynamischen Visualisierung werden die Gewinnerraten der einzelnen Neuronen in Farb- oder Helligkeitswerte umkodiert, dabei bedeutet ein hoher Zahlenwert z. B. eine helle Farbe. Das Neuron mit der höchsten Gewinnerrate wird weiß dargestellt. Neuronen mit entsprechend kleinen Raten werden z. B. dunkler visualisiert. Neuronen, die gar nicht gewonnen haben, erhalten eine schwarze Farbe auf der SOM. Durch diesen Algorithmus entstehen auf der SOM helle Flächen, die die Stützstellen der SOM repräsentieren, da die Gewichtsvektoren dieser Neuronen sehr dicht an den zu lernenden Datenbeispielen liegen. Zwischen diesen Stützstellen muß die Karte interpolieren. Damit kann dieses Visualisierungsverfahren benutzt werden, um eine Security-SOM aufzubauen. Werte für die Zielgröße und die dazugehörigen Parameter auf den hellen Flächen haben eine hohe Sicherheit (Security-Wert nahe 100%), Werte auf den dazwischenliegenden dunklen Flächen eine geringerer Sicherheit (Security-Wert wesentlich kleiner als 100%), da die Karte dort interpoliert hat. Eine mögliche Berechnung für den Security-Wert SW ist:

SW = akt. Gewinnerrate des Neurons/max. Gewinnerrate · 100% (7)

SW gibt somit in Prozent an, wie sicher die von der Karte

ermittelten Resultate sind. Die max. Gewinnerrate ist die größte Gewinnerrate die ein Neuron auf der Karte im Vergleich mit allen anderen Neuronen hat.

Als Ergebnis der obengenannten Verfahren entsteht eine strukturierte SOM-Karte, bei der es eine Zuordnung zwischen den Parametern p und den Zielgrößen z gibt. Ein SOM-Neuron M, d. h. sein Gewichtsvektor w^M , entspricht dabei genau einem Punkt in R, d. h. einer oder mehrerer Zielgrößen und den zugrundeliegenden Einflußparametern. Die Positionierung der Gewichtsvektoren w in R für alle Neuronen k der Karte wurde durch das selbstorganisierende Lernverfahren realisiert. Dabei gibt es Neuronen, die die in den Beispieldaten enthaltenen Zuordnungen zwischen Einfluß- und Zielgrößen repräsentieren, andere Neuronen geben unbekannte, nichtlinear-interpolierte Zusammenhänge an.

In der Anwendungsphase der Karte werden die interessierenden Werte der Zielgrößen oder Zielgrößenkombinationen ausgesucht und die dazugehörigen Parameterkombinationen ermittelt, indem die zu den Zielgrößenwerten gehörenden Neuronen auf der SOM selektiert werden. Diese Aus- und Anwahl kann grafisch erfolgen, da die Zielgrößen über der SOM farblich oder als Gebirge aufgetragen werden können.

Eine weitere Beschreibung des Verfahrens erfolgt nachstehend anhand der in den Zeichnungsfiguren dargestellten Ausführungsbeispiele.

Es zeigen:

Fig. 1 ein Strukturbild, das die wesentlichen Komponenten eines Systems zur Durchführung des Verfahrens zeigt,

Fig. 2 eine mögliche Gewichtsverteilung einer neuronalen Karte in einem Prozeß mit nur einem Einflußparameter und einer Zielgröße (n=2),

Fig. 3 eine neuronale farbkodierte Security-Karte zur Modellierung, mit dargestellten Stützstellen (weiße Flächen auf der Karte),

Fig. 4 eine neuronale höhenkodierte Komponenten-Karte für eine Zielgröße des Prozesses mit eingetragenem Pfad für gewünschte Werte der Zielgröße,

Fig. 5 zwei neuronale höhenkodierte Komponenten-Karte für zwei Zielgrößen A und B des Prozesses und eine daraus abgeleitete Gütefunktion C als grafische, gleichgewichtete Überlagerung der zwei Zielgrößen A und B zur grafischen Polyoptimierung,

Fig. 6 eine Prinzipschaltung zur Kombination der SOM mit feed-forward-Netzen zum Verifizieren der Ergebnisse.

Die Beschreibung des Verfahrens ist im folgenden in verschiedenen Verfahrensschritten gegliedert und durch obengenannten Abbildungen beispielhaft erläutert.

Fig. 1 zeigt beispielhaft die Struktur eines Systems zur Durchführung des Verfahrens zur Prozeßanalyse und Diagnose mit neuronaler Karte.

Verfahrensschritt 1

Aufnahme und Auswahl der Prozeßgrößen

V1.1. Aufnahme und Aufbereitung der Daten. Ein Datenerfassungs- und -aufbereitungssystem erfaßt und speichert die zur Optimierung notwendigen Prozeßgrößen.

V1.2. Ein nachfolgendes Datenanalysesystem unterzieht die ausgewählten und relevanten Daten einer Korrelationsanalyse, um die unabhängigen Parameter zu bestimmen. Danach werden die relevanten Größen skaliert und gegebenenfalls verrauscht.

V1.3. Auswertung und Analyse auf Grundlage eines Neuro-Systems.

Verfahrensschritt 2

Lernphase der Karte

V2.1. Entfaltung des selbstorganisierenden, neuronalen Netzes im Zustandsraum des Prozesses, basierend auf modifiziertem Algorithmus nach T. Kohonen.

Fig. 2 zeigt beispielhaft die Gewichtsverteilung von SOM-Neuronen in einem 2-dimensionalen Zustandsraum des Prozesses mit einem Einflußparameter (x-Achse, $w(i, 1) = w_1^i$) und einer Zielgröße (y-Achse, $w(i, 2) = w_2^i$). Die laufende Nummer i (mit $i = 1 \dots k$) gibt das ausgewählte Neuronen i des SOM-Netzes und damit den ausgewählten 2-dimensionalen Gewichtsvektoren w^i an. Die Position der Gewichtsvektoren w der Neuronen ist durch einen grauen Kreis markiert, die Nachbarschaftsbeziehungen – der zu den jeweiligen Gewichten gehörenden SOM-Neuronen – wird durch eine Linie dargestellt. Neuronen, die durch eine Linie verbunden sind, liegen auf der SOM-Karte nebeneinander. Die weißen Kreise markieren Stellen, an denen Datenvektoren vorhanden sind. Man sieht, daß der größte Teil der SOM-Neuronen dort plaziert wurde, allerdings gibt es auch Neuronen, die zwischen den Datenvektoren liegen; diese interpolieren zwischen diesen Datenbeispielen.

V2.2. Visualisierung des entfalteten Netzes als Karte durch Anwendung einer dynamischen Visualisierungsmethode, basierend auf den jeweiligen Gewinnraten der Neuronen oder durch eine statische Visualisierungsmethode nach dem Prinzip der UMatrix-Methode.

Fig. 3 zeigt – hier in Schwarz/Weiß Darstellung – beispielhaft die strukturierte und farbkodierte, neuronale Security-Karte für den Herstellungsprozeß eines Maschinenteiles. Es stellt die Projektion und Visualisierung von 30 Prozeßgrößen auf einer neuronalen Karte mit $20 \cdot 30$ Neuronen dar. In Fig. 3 gibt die x-Achse die Anzahl der Neuronen in x-Richtung, die y-Achse die Anzahl der Neuronen in y-Richtung an. Die Prozeßgrößen bestehen in diesem Beispiel aus den Zielgrößen Drehmoment und Lagertemperatur und 28 Einflußparametern.

Die hellen Flächen sind die ermittelten Stützstellen; Werte der Zielgröße Drehmoment, die an diesen Stellen berechnet werden, sind sehr sicher. Die dunklen Gebiete auf der Karte bilden die Hülle zwischen diesen Stützstellen, da sie Neuronen darstellen, die im Zustandsraum zwischen den gelernten Daten angeordnet sind.

Obwohl in diesem Beispiel jeder Prozeßvektor durch 28 unabhängige Parameter (z. B. Bohrungsdurchmesser, Lagerlänge, Materialrauhigkeit) und 2 Zielgrößen (Drehmoment und Lagertemperatur) bestimmt wird und damit 30-dimensional ist, kann durch das Verfahren eine topologieerhaltende Projektion auf nur zwei Dimensionen, die SOM-Karte, durchgeführt werden. Die Anzahl der gleichzeitig ausgewerteten Größen ist nicht auf 30 beschränkt, sie kann wesentlich höher sein. Entscheidend ist die topologieerhaltende Projektion. Im obengenannten Beispiel stellt jede Zielgröße ein Gebirge auf dem 28-dimensionalen Parameterraum dar und obwohl dieses Gebirge existiert, ist es mit herkömmlichen Methoden nicht zugänglich. Die Anzahl der Neuronen hängt a-priori nicht von der Anzahl der Prozeßgrößen ab. Die Neuronenanzahl sollte so groß wie möglich gewählt werden, sie wird in der Regel nur durch die Rechenleistung des zugrunde liegenden Neuro-Systems beschränkt.

Durch die SOM-Karte gelingt es, die grundlegenden topologischen Zusammenhänge der Gütefunktionen in dem hochdimensionalen Zustandsraum beizubehalten und auf 2 bzw. 3 Dimensionen zu projizieren und zu visualisieren. Damit bekommt ein Beobachter einen Einblick in die Struktur des hochdimensionalen Gebirges der Gütefunktion oder –

wie im Beispiel – einer Zielgröße.

Wie eingeführt, hat jedes Neuron M auf der Karte einen Gewichtsvektor w^M mit n Elementen, mit $w^M = (w_1^M, w_2^M, w_3^M, \dots, w_n^M)$. Mit w_i wird die i-te Komponente eines Gewichtsvektors bezeichnet. Eine Komponentenkarte entsteht dadurch, daß man die Werte w_i^k aller Neuronen k der Karte über der SOM visualisiert. Da den Gewichtswerten jedes Neurons eindeutig der Wert der zugrundeliegenden Prozeßgrößen zugeordnet werden kann, erhält man durch diese Darstellung eine Verteilung der jeweiligen Zielgröße j über den gesamten Zustandsraum des Prozesses. Bei der Umrechnung von dem Gewichtswert in den realen Zielgrößenwert muß gegebenenfalls die Skalierung beachtet werden. Die Visualisierung der so erhaltenen Werte über der SOM erfolgt im Falle des Gebirges höhenkodiert. Genauso gut ist es möglich, den Zielgrößenwert in einen Farbwert umzurechnen und auf der SOM aufzutragen. Durch eine entsprechende Skala kann damit der Wert der Zielgröße abgelesen werden.

Fig. 4 zeigt beispielhaft das auf eine SOM projizierte höhenkodierte Gebirge der Zielgröße Drehmoment. Dieses Bild entsteht durch Aufbau einer höhenkodierten Komponentenkarte für die Komponente Drehmoment. Die x- und y-Achse der Karte stellt dabei den Definitionsbereich der Zielgröße, die z-Achse stellt den Wertebereich der Zielgröße dar, d. h. für jedes SOM-Neuron wird der dazugehörige Zielgrößenwert berechnet und in einem 3-dimensionalen Plot aufgetragen. Es ist zu beachten, daß der Definitionsbereich nicht – wie gewöhnlich – aus physikalischen Werten zweier Einflußparameter, sondern gleichzeitig aus allen 28 Einflußgrößen besteht. Diese 28 Einflußparameter wurden zusammen mit den Zielgrößen topologieerhaltend auf die Karte abgebildet.

Verfahrensschritt 3

Anwendung – Optimierung einer Zielgröße des Prozesses

Da die realen Werte der Zielgröße als Gebirge über der SOM aufgetragen werden können, kann man einfach die gewünschten Werte auf dem Gebirge markieren und die dazugehörigen Parameterkombinationen ausgeben lassen. Dabei lassen sich die Zielwerte z. B. wie Höhenlinien auf dem Gebirge auftragen oder es ist möglich, auf dem Zielgrößengebirge einen Pfad P von zulässigen Werten zu definieren und für diesen Pfad alle dazugehörigen Parameterkombinationen von der SOM berechnen zu lassen. Fig. 4 stellt einen solchen Pfad – als dicke schwarze Linie – auf dem Gebirge eines Drehmomentes dar. Der so definierte Zielwert liegt normiert bei 700. Desweiteren ist es möglich, zulässige Bereiche $\Delta z = z_{MAX} - z_{MIN}$ für die Zielgröße auszuwählen. Dieser Bereich entspricht in den höhenkodierten Darstellungen einer Toleranzbreite in z-Richtung, dieser Bereich ist damit wie ein zweifacher Schnitt parallel zur x,y-Ebene, einmal in der Höhe z_{MAX} und einmal bei z_{MIN} zu betrachten.

Die Grundidee ist, daß ausgehend von einem gewünschten Zielwert auf der SOM die zugrundeliegenden Optimierungsparameter ermittelt werden können.

Die Rückrechnung von der Zielgröße zu den Parametereinstellungen ist deshalb möglich, weil jedem Wert der Zielgröße im Definitionsbereich ein SOM-Neuron zugeordnet werden kann. Z.B. wird für den in Fig. 4 geforderten Wert Z2 das Neuron M auf der Karte angewählt. Durch diese Auswahl des Neurons M werden die jeweiligen Gewichtsvektoren w^M ausgegeben und in die realen Werte der Prozeßparameter umgerechnet. Dadurch lassen sich genau die gesuchten Parameterkonfigurationen p_1, p_2, \dots, p_N ermitteln, die eben zu dem gewünschten Zielwert geführt haben. Diese ge-

fundenen Parameterkombinationen können durch ein nachgeschaltetes Verfahren auf Plausibilität untersucht werden und, wenn gefordert, an den Prozeß on-line oder off-line ausgegeben werden.

Dadurch, daß das Gebirge der Zielgröße grafisch visualisiert ist, ist es ohne weiteres möglich, auch die Minima oder Maxima der Zielgröße auszuwählen; man sieht weiterhin auch für den Fall, daß es mehrere Bereiche mit den gleichen Werten der Zielgröße gibt, die Einbettung dieses Wertes in den Kontext anderer Werte. Damit lassen sich robuste und fehlertolerante Optimierungen ermitteln, denn man wird im Zweifelsfall nicht den Zielwert an einem steilen Hang des Gebirges auswählen, z. B. Z1 in Fig. 4, wenn es noch eine Alternative auf einem Plateau gibt, z. B. Z2 in Fig. 4. Beide Punkte entsprechen einem gleichen numerischen Wert von 800 für die Zielgröße, siehe z-Achse, Z2 ist aber wesentlich stabiler. Man findet damit für eine geforderte Zielgröße Parameterkombinationen, die zusätzlich noch relativ robust gegenüber kleinen Änderungen in den Parameterwerten sind.

Die eigentliche Optimierung geschieht demnach durch die Auswahl des gewünschten Zielwertes z auf dem SOM-Gebirge und die anschließende Berechnung der zugrundeliegenden Parameterkombinationen p_1, \dots, p_N durch die SOM-Karte. Die SOM-Karte kann damit wie ein inverses Modell f verwendet werden, mit $(p_1, \dots, p_N) = f(z)$.

Die Auswahl des gewünschten Zielwertes geschieht weiterhin im Kontext des gesamten Zielgrößengebirges. Damit lassen sich für den Fall, daß es mehrere ähnliche Zielwerte gibt, diejenigen auffinden, die im Hinblick auf andere Eigenschaften am günstigsten für die Optimierungsaufgabe sind. Am günstigsten könnte z. B. am robustesten sein, da die Zielgröße auf einem Plateau des Gebirges liegt.

Auch die Suche nach Minima oder Maxima ist sehr einfach, da man dem Gebirge ohne weiteres ansieht, wo ein Berg MAX oder ein Tal MIN ist. Genau an diesen Stellen hat man ein Minimum oder ein Maximum gefunden und kann sich die dazugehörige Parameterkombination ausgeben lassen. Entscheidend ist, daß die Karte interpolieren kann und damit Zielwerte dargestellt und ausgewählt werden können, die in den Optimierungsbeispielen nicht vorhanden waren. Im Zusammenhang mit der kontextabhängigen Auswahl, lassen sich damit robuste Optimierungen realisieren, die ohne zugrundeliegendes mathematisches Modell durchgeführt werden können und die damit universell anwendbar sind.

Benutzt man weiterhin die in Fig. 3 dargestellten Ergebnisse der Security-SOM über die Sicherheit der SOM-Berechnung, kann man auf dem Gebirge auch noch zwischen sicheren und unsicheren Optimierungsergebnissen unterscheiden. Damit kann man eine Aussage darüber erhalten, wie wahrscheinlich der Verlauf des SOM-Gebirges mit der Realität im Zustandsraum P übereinstimmt. Helle Ausprägungen des Gebirges haben eine hohe Sicherheit, dunkle Darstellungen entsprechen unsicheren Gebirgsverläufen.

Die Plateaus der Zielgröße sind besonders dann interessant, wenn robuste Optimierungen gesucht werden, Optimierungen also, bei denen Änderungen in den Einflußparametern nicht zu gravierenden Änderungen der Zielgröße führen. Gerade für diese sog. fehlertolerante Optimierung, sind diese Plateaus eine völlig neuartige Möglichkeit der Parameterauswahl. Sie ist generell anwendbar, nichtlinear und auch intuitiv verstehbar, denn ein Zielwert auf einem Plateau bedeutet, daß dieser bei kleineren Änderungen von Werten der Einflußgrößen auf dem Plateau wandern wird, daß sich damit sein numerischen Wert nur sehr wenig ändert. Andererseits reagiert ein Zielgrößenwert, z. B. z_M , dessen zugehöriges Neuron M auf dem Hang eines Gebirges

liegt sehr empfindlich auf Änderungen in den Einflußgrößen p_1, \dots, p_N . Kleine Änderungen von p_1, \dots, p_N bewirken auch hier nur, daß Neuronen um das ursprüngliche Gewinnerneuron M ausgewählt werden, die Zielgrößenwerte für diese benachbarten Neuronen können sich aber groß verändern, da ihr Zielgrößenwert auf einem Gebirgshang liegt.

Weiterhin ist es möglich, aufbauend auf den Werten einer Zielgröße eine Gütefunktion Q zu berechnen und diese anstatt der Zielgröße selbst darzustellen.

Verfahrensschritt 4

Anwendung – Polyoptimierung mehrerer Zielgrößen

Für den Fall, daß mit der Karte mehrere Zielgrößen L trainiert wurden ($2 \leq L \leq N$), können diese Zielgrößen miteinander ausgewertet werden, indem eine Gütefunktion G eingeführt wird, die aus mehreren Zielgrößen berechnet wird. Da jede Zielgröße als Gebirge über der Karte dargestellt werden kann, erhält man durch Überlagerung dieser Gebirge ein völlig neues Gebirge, mit veränderten Eigenschaften. Minima und Maxima auf diesem neuen Gebirge geben an, daß hier z. B. der Zielgrößenvektor aller Zielgrößen minimal oder maximal ist. Dadurch lassen sich die besten Kompromisse für die überlagerten Zielgrößen finden.

Fig. 5 zeigt ein Beispiel, bei dem die Zielgröße Drehmoment (Teil A, Fig. 5) und Lagertemperatur (Teil B, Fig. 5) überlagert wurden. Das entstandene Gebirge GEB ist in Teil C von Fig. 5 dargestellt. Für diese Überlagerung gibt es mehrere Möglichkeiten: Zum einen kann eine einfache additive Überlagerung der Zielgrößen realisiert werden, indem für jedes Neuron M auf der Karte die L Zielgrößenwerte, die in den Gewichtsvektor w eines Neurons M gespeichert sind, aufaddiert werden:

$$\text{wert}_G^M = \text{wert}_{z_1}^M + \text{wert}_{z_2}^M + \dots + \text{wert}_{z_j}^M + \dots + \text{wert}_{z_L}^M \quad (8)$$

mit $M = 1 \dots k$ und wert_G^M als Wert der neuen – aus dem Vektor berechneten – Gütefunktion G an der Stelle M der Karte und $\text{wert}_{z_j}^M$ als Ausgangswert der j -ten Zielgröße an der Stelle M der Karte. In Teil C von Fig. 5 wurde das Ergebnis einer einfachen additiven Überlagerung der zwei Zielgrößen Drehmoment und Lagertemperatur dargestellt.

Andererseits ist es auch möglich, bestimmte Zielgrößen mehr als andere zu wichten. Das geschieht dadurch, daß eine gewichtete Überlagerung der Zielgrößen z_1, z_2, \dots, z_L durchgeführt wird.

$$\text{wert}_G^M = a_1 \cdot \text{wert}_{z_1}^M + a_2 \cdot \text{wert}_{z_2}^M + \dots + a_L \cdot \text{wert}_{z_L}^M \quad (9)$$

Allgemein läßt sich damit jede gewünschte Gütefunktion aus den Zielgrößen berechnen, mit

$$G = f(z_1, z_2, \dots, z_L). \quad (10)$$

Die Funktion f kann jede beliebige mathematische Funktion sein. Nach Berechnung der neuen Werte wert_G an allen Stellen auf der Karte, d. h. für alle Neuronen, und gegebenenfalls nach anschließender Skalierung wird das Ergebnis wieder als Gebirge GEB über der Karte aufgetragen. Auf diese Weise läßt sich eine universelle Optimierung gleichzeitig für mehrere Zielgrößen durchführen, denn ausgehend von diesem neuen Gebirge lassen sich für jeden gewünschten Zielwert die zugrundeliegenden Parameterkombinationen berechnen, nach $(p_1 \dots p_N) = g(f(z_1, z_2, \dots, z_L))$, mit g entspricht einer inversen Abbildung zwischen den Parametern und den Zielgrößen.

tern P und den Zielgrößen Z. Die Funktion g muß im mathematischen Sinne nicht existieren, sie wurde durch das SOM-Verfahren während der Entfaltung ermittelt.

Da auch die Gütefunktion G stets als Gebirge GEB dargestellt werden kann, stellt das Verfahren eine Erweiterung gegenwärtiger Methoden der Mehrgrößenoptimierung (Polyoptimum) in Forschung, Entwicklung und Konstruktion und in industriellen und chemischen Prozessen dar. Der Vorteil ist, daß ein Minimum oder Maximum auch der Kompromißlösung, einfach grafisch ermittelt wird und die jeweils dazugehörigen Parameterkombinationen ausgegeben werden können. Der Anwender muß nur die Zielgrößen seiner Wahl miteinander verknüpfen und die entstandene Gütefunktion wird als neues Gebirge dargestellt.

Verfahrensschritt 5

Anwendung – Verifizierung der Analyseergebnisse

Zur Verifizierung der Parameteroptimierung wird ein neuronales feed-forward-Netz verwendet, welches den gleichen Versuchsprozeß wie die SOM modelliert hat. Im Unterschied zur SOM werden bei den neuronalen feed-forward-Netzen die Ein- und Ausgänge getrennt behandelt. Alle Parameter, d. h. alle Einflußgrößen für die Optimierung, werden an die Netzeingänge gelegt. Alle Zielgrößen werden mit den Netzausgängen verbunden. Durch Anwendung verschiedener neuronaler Lernverfahren erlernt das Netz eine Approximation zwischen seinen Ein- und Ausgängen. Dadurch kann man den gesamten Prozeß modellieren.

Ein feed-forward-Netz ist ein eingeführter Begriff für eine Klasse von Netzen, deren Neuronen geschichtet angeordnet sind und bei denen es keine Rückführungen zwischen Neuronen einer höheren zu Neuronen einer niedrigeren Schicht gibt; siehe A. Zell, "Simulation Neuronaler Netze", Addison Wesley Verlag, 1. Auflage, 1994, Seiten 76 bis 78. Diese Netze sind geeignet, statische und dynamische Modelle zwischen gewählten Größen aufzubauen. Ausgehend von einem neuronalen Modell, das zwischen den Versuchsparametern p_1, p_2, \dots, p_N und der Zielgröße z_j aufgebaut wurde, kann dieses Modell zum Verifizieren der Parameteroptimierung verwendet werden.

Fig. 6 verdeutlicht das Prinzip. Dabei muß das feed-forward-Netzmodell mit den gleichen Versuchsbeispielen wie die SOM trainiert werden. Als neuronales feedforward-Netzmodell wird ein Security-RBF-Modell aufgebaut. Ein Security-Netz ist ein Netz, daß zusätzlich zu seinen Modellausgangswert angibt, wie sicher dieser Modellausgangswert ist. Diese Sicherheit kann z. B. durch einen Zahlenwert von 0 ... 100% ausgegeben werden: 0% bedeutet dabei keine Sicherheit, 100% bedeutet, daß das Netz-Ergebnis ganz sicher ist.

Bei Verwendung geeigneter, lokal approximierender Netze, z. B. RBF-Netze (siehe A. Zell, Seite 225 bis 239), ist es möglich, solche Security-Netze aufzubauen. RBF-Netze sind 3-schichtige Netzwerke, deren verdeckte Hidden-Schicht aus Neuronen besteht, die eine gaußförmige Aktivierungsfunktion haben. Ihre maximale Ausgangsaktivität ist auf einen kleinen Eingangsbereich beschränkt. Diese Neuronen sind damit nur lokal für einen kleinen n-dimensionalen, hyperkugligen Eingangsbereich sensitiv. Liegt der Eingangsvektor in diesem Bereich, hat das jeweilige Hidden-Neuron seine maximale Ausgangsaktivität, die anderen Hidden-Neuronen haben eine entsprechend geringere Aktivität (siehe A. Zell).

Durch Berechnung eines Abstandsmaßes zwischen angelegtem Eingangsvektor und dem Hidden-Neuron mit maximaler Aktivität, erhält man ein Maß dafür, ob das Netz ge-

genwärtig in einem Bereich angewendet wird, in dem es nicht trainiert wurde. Ist dieses Abstandsmaßes groß, heißt das, daß der Eingangsvektor sehr weit von den Stützstellen des RBF-Netzes entfernt ist. Damit ist das Ergebnis des Netzes relativ unsicher, denn neuronale Netze interpolieren zwischen den gelernten Stützstellen. Diese Interpolation ist um so unsicherer, je weiter der Wert von den nächsten Stützstellen entfernt ist. Befindet sich der aktuelle Ausgangswert im Extrapolationsraum, ist das Ergebnis noch unsicherer.

Ein Beispiel für die Berechnung des Security Wertes SW_{RBF} ist:

$$SW_{RBF} [\%] = activity_{WINNER} [\%] - MSE_{Learn} [\%] \quad (11)$$

wobei SW_{RBF} der Sicherheitswert der Netzantwort in Prozent ist, $activity_{WINNER}$ gibt die Aktivitätsantwort des Gewinnerneurons auf einen Eingangswert multipliziert mit 100% an und MSE_{Learn} ist der mittlere Lernfehler des Netzes, aufgenommen während des Trainings. Damit kann der Securitywert nicht größer als die Genauigkeit während des Lernvorganges sein. Der Aktivitätswert ist ein Maß für den Abstand des Eingangsvektor zu der am nächsten gelegenen gelernten Stützstelle. Dadurch entsteht ein Security-RBF-Netz, das für jede Simulation nicht nur das Simulationsergebnis, sondern auch eine Sicherheit des Ergebnisses berechnet und ausgibt.

V5.1. Die für einen ausgewählten Zielgrößenwert z_j ermittelten optimalen Testparameter $p_{1, opt}, \dots, p_{N, opt}$ in Fig. 6 werden den Security-RBF-Netz angeboten und die dazugehörige Zielgröße $z_{j, TEST}$ ermittelt. Stimmt $z_{j, TEST}$ mit z_j annähernd überein, ist also die Differenz $DIFF = |z_j - z_{j, TEST}|$ kleiner als ein vorgegebener Wert OK, sind die Ergebnisse der SOM verifiziert, da die beiden grundlegend-verschiedenen Neuromodelle gleiche Modellergebnisse berechnet haben. Damit hat man die N verifizierten, optimalen Parameter $p_{1, opt}, \dots, p_{N, opt}$ für den gewünschten Zielwert bestätigt. Im Negativfall muß die Auswahl $z_{j, TEST}$ iterativ neu erfolgen bzw. die SOM- oder das RBF-Modell neu trainiert werden.

Das heißt, mit der SOM sucht man ausgehend von dem gewünschten Zielwert z_j die dazugehörigen Parameter und mit dem Security-RBF-Netz verifiziert man diese Ergebnisse, da die jeweiligen Parameter wieder zu dem auf der SOM ausgewählten Zielwert führen müssen.

V5.2. Im Falle der Polyoptimierung müssen die mit der SOM ausgesuchten Parameterkombinationen für einen Kompromißwert (z. B. Minimum oder Maximum) durch das Security-RBF-Netz annähernd zu den – auf der SOM ausgewählten – einzelnen Werten jeder Zielgröße führen.

Durch die Kombination des zielorientierten Ansatzes der SOM mit den Verifikationsmöglichkeiten eines feed-forward-Modells lassen sich die gegenwärtigen Methoden der Optimierung erweitern, da die hier beschriebenen Verfahren nichtlinear und universell, d. h. ohne zugrundeliegendes mathematisches Modell, anwendbar sind. Insbesondere können Optimierungen ermittelt werden, die robust gegenüber Änderungen in den Optimierungsparametern p sind.

Ein weiterer Vorteil des Verfahrens liegt in dem geringen Engineering-Aufwand. Da das Lernen und Strukturieren der Karten vollständig durch das Verfahren erbracht wird, entfällt aufwendiges Engineering oder Parametrisieren von Regeln, Modellen, oder Differentialgleichungen. Das Engineering ist integraler Bestandteil des Verfahrens selbst.

Patentansprüche

1. Verfahren zur Lösung von Optimierungsaufgaben in Forschung, Entwicklung und Konstruktion und zum Optimieren von technischen oder chemischen Produk-

ten und von Prozessen zur Herstellung dieser Produkte, bei dem alle für die Optimierung relevanten Größen zusammengefaßt und durch eine neuronale Analyse auf der Grundlage selbstorganisierender Karten, sogenannter SOM, in Beziehung zueinander ausgewertet werden, indem eine topologieerhaltende, nichtlineare Projektion von Daten der relevanten Parameter und die dazugehörigen Zielgrößen auf eine mehrdimensionale SOM realisiert wird, wobei zur Optimierung einer Zielgröße diese als SOM-Komponentenkarte entweder höhenkodiert in Form eines Gebirges (GEB) oder farbkodiert visualisiert wird und nach einer Auswahl von Werten der Zielgröße (Z1, Z2) auf der SOM-Komponentenkarte die zugrundeliegenden Parameterkombinationen berechnet und ausgegeben werden.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß Produkte aus dem Bereich Maschinen- und Fahrzeugbau, des Anlagenbaus, der Feinwerktechnik, der Chemischen und der Gentechnische Industrie sowie Werkstoffe optimiert werden.

3. Verfahren nach Anspruch 2, dadurch gekennzeichnet, daß Produkte des Anlagenbaus Komponenten der Kraftwerkstechnik sind.

4. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß im Fall höhenkodierter Darstellung die gewünschten Werte einer Zielgröße durch eine oder mehrere Höhenlinien auf dem Zielgröbengebirge oder durch Festlegung eines Pfades (P), eines Bereiches oder eines Plateaus auf dem Gebirge ausgewählt werden.

5. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß das Zielgröbengebirge mit den Sicherheitswerten einer Security-SOM verglichen werden, um die Bereiche auf der SOM zu identifizieren, die die sichersten Werte für die Optimierung repräsentieren.

6. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß mehrere Zielgrößen nach Art einer Polyoptimierung gleichzeitig und zusammenhängend optimiert werden, indem die ausgewählten Zielgrößen durch eine beliebige mathematische Funktion miteinander verknüpft werden und die resultierende Gütefunktion (G) als Gebirge (GEB) über der SOM aufgetragen wird, und ausgehend von dem neuen Gebirge die zugrundeliegenden Parameterwerte ermittelt werden.

7. Verfahren nach Anspruch 4 oder 6, dadurch gekennzeichnet, daß auf dem überlagerten Gebirge (GEB) Höhenlinien, Pfade, Bereiche oder Plateaus eingetragen werden und die dazugehörigen Parameterkombinationen ermitteln werden.

8. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß seine Durchführung in nachstehenden Schritten erfolgt:

- a) Aufnahme und Auswahl der Versuchsparameter und Aufbereitung der Daten,
- b) Entfaltung des selbstorganisierenden, neuronalen Netzes im Zustandsraum des Prozesses, basierend auf dem SOM-Algorithmus, wobei die Werte der relevanten Versuchsparameter verwendet werden,
- c) Darstellung des entfalteten Netzes als neuronale Karte,
- d) gegebenenfalls Projektion des hochdimensionalen Parameterraums auf die neuronale Karte, basierend auf der UMatrix-Methode oder mittels sogenannter winner-takes-all Algorithmen, mit anschließender Visualisierung der aufsummierten

Gewinnerraten der einzelnen Neuronen auf der Karte zum Aufbau einer Security-SOM,

- e) Darstellung der Zielgrößen höhen- oder farbkodiert auf der SOM,
- f) im Fall höhenkodierter Darstellung: Festlegung von Höhenlinien, Pfaden oder Bereichen auf der SOM zur Auswahl der gewünschten Werte der Zielgrößen,
- g) Berechnung der den gewünschten Zielwerten zugrundeliegenden Parameterkombinationen,
- h) gegebenenfalls gewichtete oder ungewichtete Überlagerung der Gebirge mehrerer Zielgrößen und Festlegung der gewünschten Bereiche auf der neu entstandenen Zielgrößen- oder Gütefunktionsverteilung.

9. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Ergebnisse, die mittels der SOM erhalten wurden, durch ein nachgeschaltetes Security-RBF-Modell verifiziert werden.

Hierzu 5 Seite(n) Zeichnungen

- Leerseite -

THIS PAGE BLANK (USPTO)

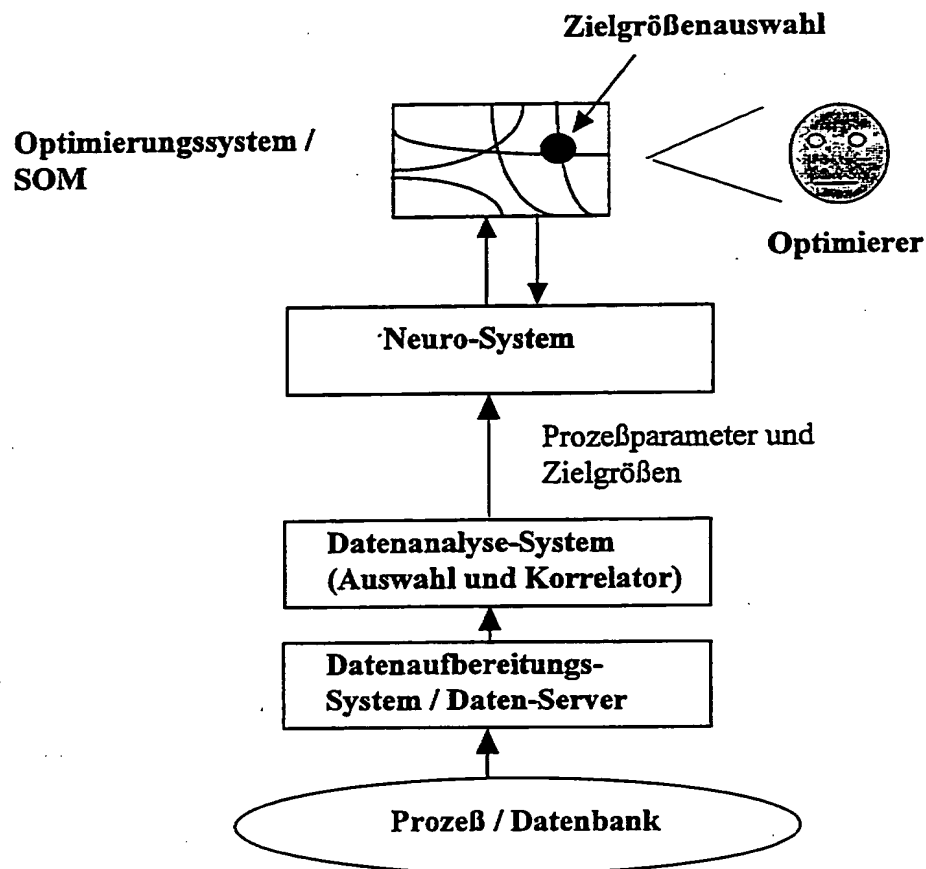


Fig. 1

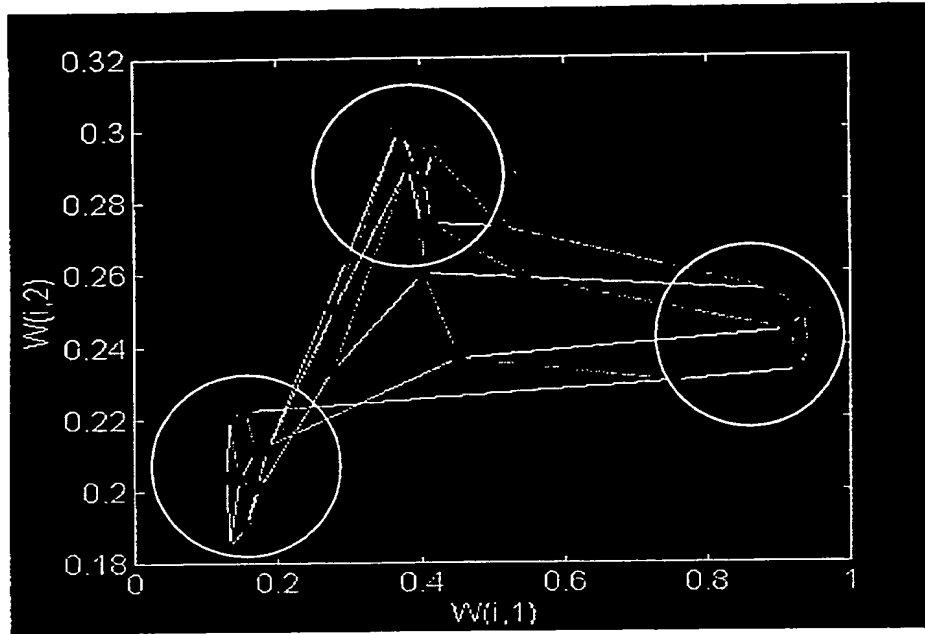


Fig. 2

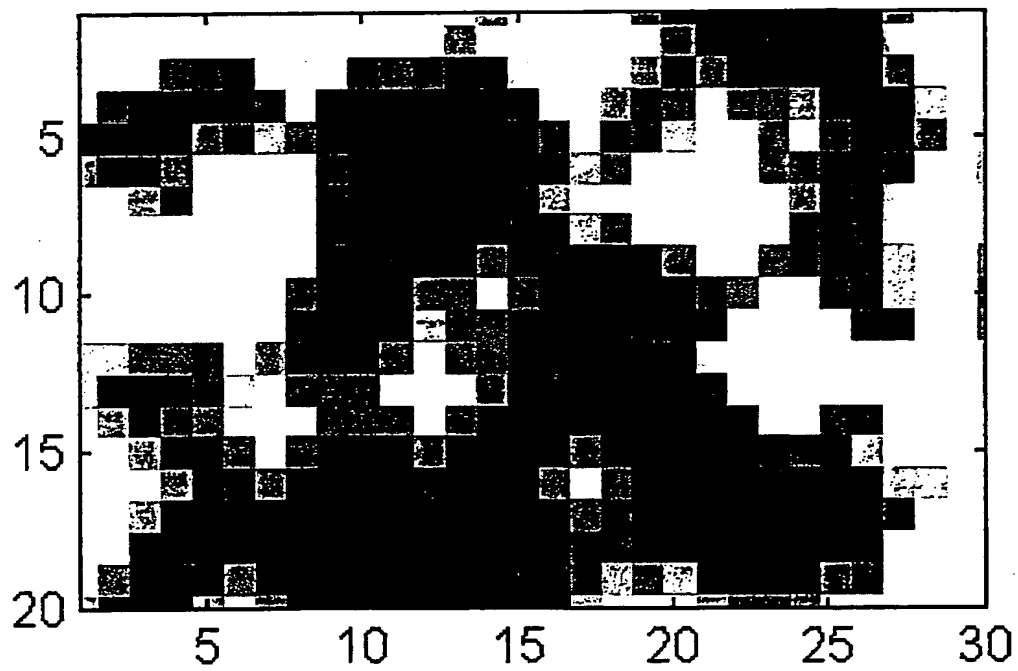


Fig. 3

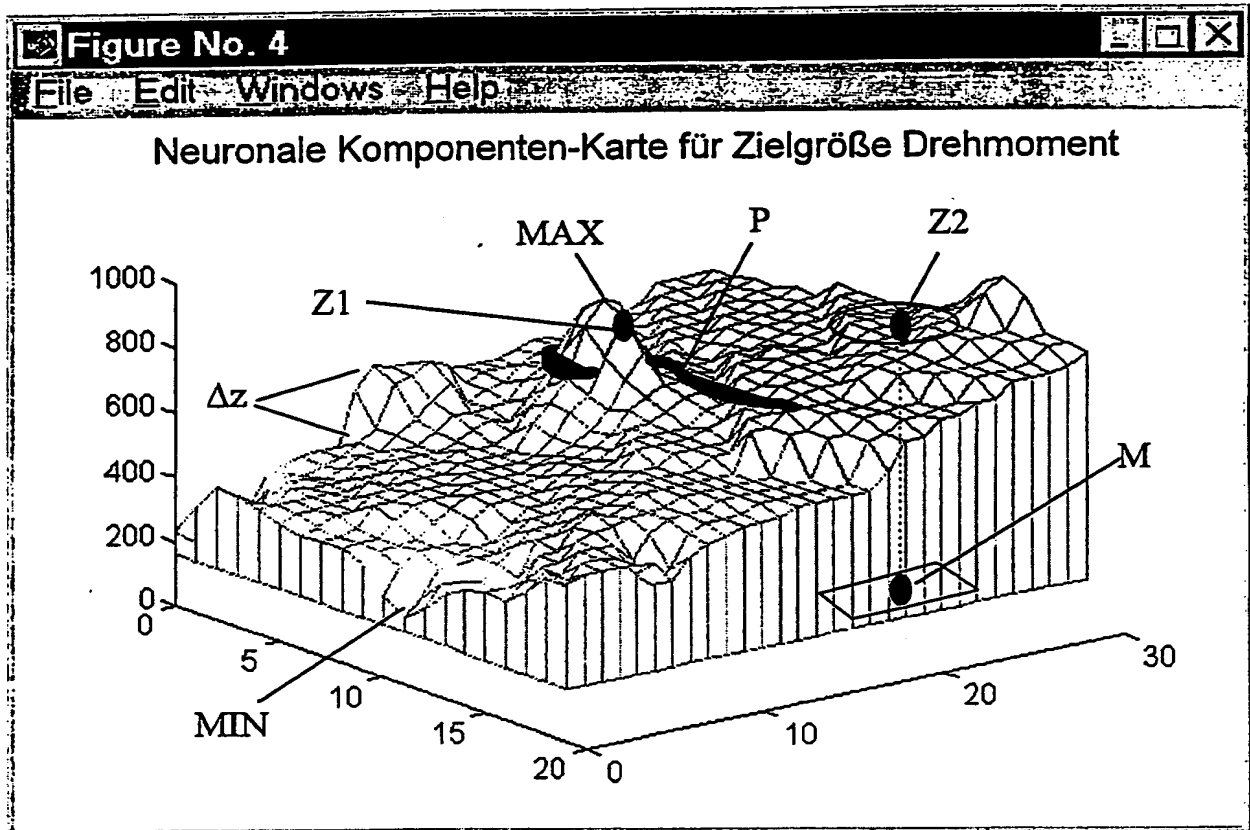
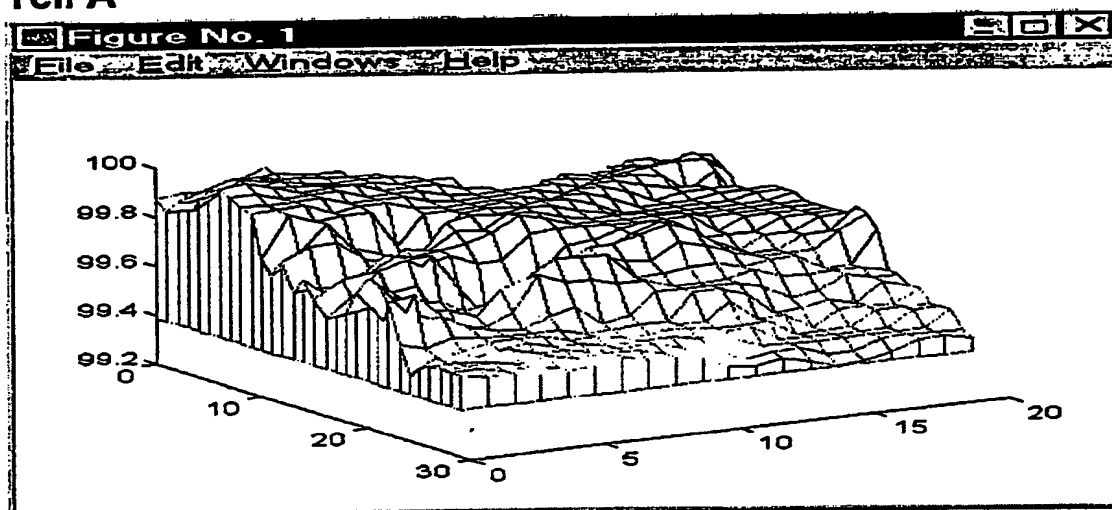
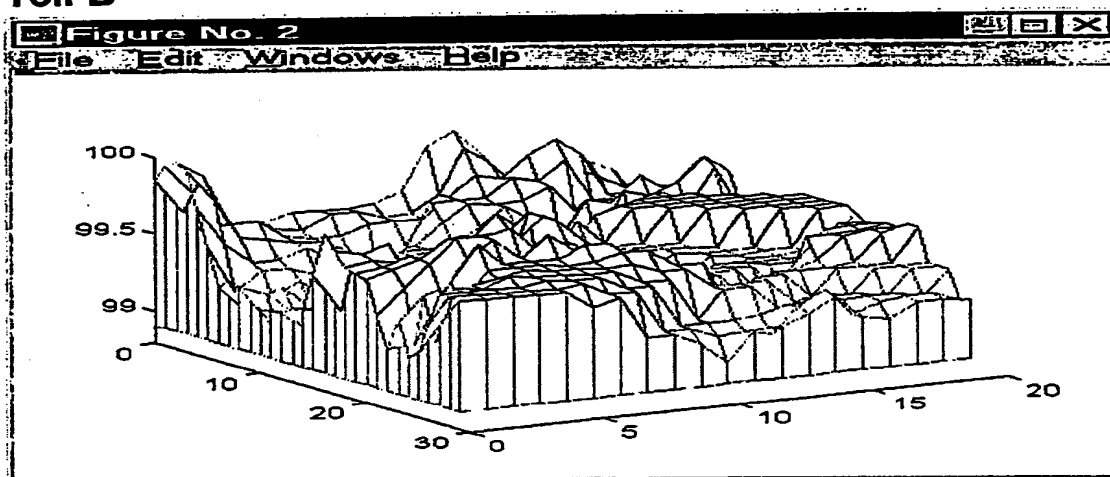


Fig. 4

Teil A



Teil B



Teil C

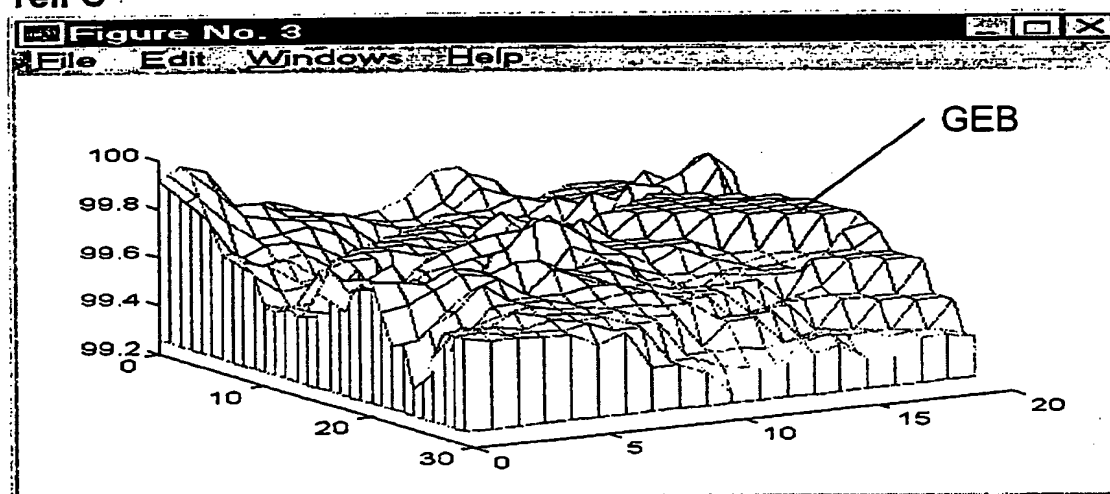


Fig. 5

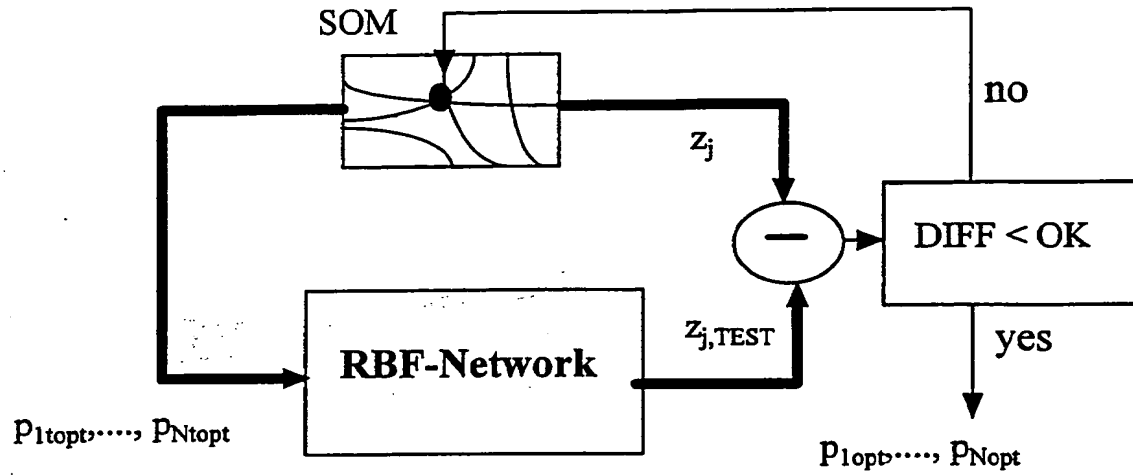


Fig. 6